Automatic Generation of Test Infrastructures for Analog Integrated Circuits by Controllability and Observability Co-optimization

Anthony Coyette[a], Baris Esen[a], Wim Dobbelaere[b], Ronny Vanhooren[b], Georges Gielen[a]

[a] *Department of Electrical Engineering, KU Leuven, Leuven, Belgium*
[b] *On Semiconductor Belgium, Oudenaarde, Belgium*

Abstract

This paper presents a method to address the automatic testing of analog IC's for catastrophic defects. Based on Design-for-Testability building blocks offering extra controllability and extra observability, a test infrastructure is generated for a targeted circuit. The selection of the extra blocks and their insertion into the circuit is done automatically by a workflow based on DC simulations and optimization algorithms. Adopting a defect-oriented methodology, this approach maximizes the fault coverage while minimizing the silicon area overhead and test time. The proposed method is applied to two industrial circuits in order to generate optimal test infrastructures combining controllability and observability. These case studies show that, with a silicon area overhead of less than 10%, a fault coverage of 94.1% can be reached.

Keywords: `Analog ATPG`, `Automatic Testing`, `Controllability and Observability`, `Design-for-Testability`

1. Introduction

The testing of analog Integrated Circuits (IC's) has been a center of focus for many years. While advances in this field have been made, the challenge of testing analog IC's in an automated way still remains. At the same time, electronics continues to extend its presence in various domains and leads engineers to face new quality requirements. For instance, the automotive industry combines nowadays an average of 400 IC's per vehicle and intends to increase this number in the future with, for instance, the expected arrival of self-driving cars. Since the defect probabilities of all components of a system multiply with each other, the requirement on each component increases and defect levels below the part-per-million (ppm) are desired. These tightening quality requirements combined with a shortening time-to-market put pressure on IC designers and manufacturers. Therefore, an advancement is needed in the testing of analog IC's and its automation.

The testing of digital IC's has known an automation of its process and an improvement leading to defect levels under the ppm. This success was made possible by using a fault-based approach and by the appearance of automated algorithms such as PODEM [1] or FAN [2]. Furthermore, a generic Design-for-Testability (DfT) approach based on flip-flops connected in a scan chain was developed and enabled the automatic utilization of these algorithms.

In comparison, the field of analog IC testing has not yet known this same level of automation. Currently, research has delivered Built-In Self-Test (BIST) structures which focus on the testing of specific types of circuits (ADC/DAC, PLL, etc.)[3, 4], but no automated tool is at the designers' disposal to enable the testing of analog circuits like the digital scan chain does in digital IC's.

The problem of testing analog circuits in a generic way has been addressed in works such as [5, 6] where analog scan chains are proposed. In the same way as in digital scan chains, voltages can be scanned through sample-and-hold (S/H) circuits and imposed on node voltages. Similarly, node voltages can be read and scanned out of the chips by chains of S/H circuits. While these methods tackle the testing of analog circuits in a generic approach, they suffer from several drawbacks. The parasitics imposed on the probed nodes by the analog buses have been criticized. Also, the forcing of voltages on internal nodes requires the presence of multiple buffers. The whole approach requires a significant silicon area overhead.

In this paper, a method is proposed to automatically generate a DfT infrastructure in order to test analog IC's. The presented method combines small building blocks offering extra controllability and extra observability to the circuit under test (CUT). This co-optimization of controllability and observability offers an alternative to the analog scan chains without requiring a large silicon area. This paper comes as an extension to a work previously presented [7]. A more detailed description of that implementation is given here and new considerations are implemented.

In Section 2, the defect-oriented methodology enabling the method is summarized. Then, the building blocks forming the basis of the method are presented in Section 3. In Section 4, the workflow of the method is presented from

*Corresponding author:
Email address: `acoyette@esat.kuleuven.be` (Anthony Coyette)

the DC simulations to the combination of the building blocks through an optimization system. In Section 5, two industrial circuits are taken as case studies and simulations results are shown. Finally, conclusions are drawn in Section 6.

2. Defect-Oriented Method

In the defect-oriented approach, the physical defects which can occur in IC's are considered and are simulated with fault models [8–10]. Defects can be distinguished into two categories: catastrophic and parametric defects. The former emerge from a problem during the manufacturing process such as an over- or under-etching, the presence of a dust particle, etc. They cause a modification of the designed topology i.e. a short circuit or an open circuit. The latter emerge from an imperfect control of the process, voltage and temperature (PVT) conditions. These PVT variations cause variability among the produced IC's, resulting in some IC's laying outside of the targeted specifications. In the scope of this work, the focus is put on automotive applications where the used technologies are typically above 100 nm and hence are mastered well enough to apply a 6σ design flow. Therefore, parametric defects are under control and only catastrophic defects are considered in this work.

The modeling of the defects is done at schematic level with models from literature [11]: the 5-fault model for the MOSFETs and the 6-fault model for the bipolar transistors. The application of these models on a circuit C_0 results in a list of faults F_1,...,F_n. Then, one by one, the faults from the generated list are separately inserted into the original circuit C_0. The insertion of a fault F_i into the circuit C_0 leads to the faulty circuit C_i. The transistor-level simulation of the circuit C_i allows to estimate the effects of the fault on the circuit. The method developed in the following makes use of results coming from these DC simulations in the presence of process variations.

3. Controllability and Observability Structures

In this section, the concepts of controllability and observability are introduced as defined in [12]. The controllability is defined as the relative difficulty of setting a node to a specific value. The observability is defined as the relative difficulty of measuring the signal value at a node. The combination of these two concepts forms the basis for an optimized way to test an analog integrated circuit. The main idea is to control a circuit and lead it into a region of operation where a different behavior can be observed between a faulty circuit and a good circuit.

In the following, two techniques are presented in order to enhance the controllability and the observability of analog IC's. The controllability is enhanced by using the Topology Modification method introduced in [9]. The observability is enhanced by using the Local Detection and Transmission Systems introduced in [11]. The combination of both techniques provides a simplification of the test infrastructure as compared to analog scan chains. As a result, DfT Control and Observation Structures (COS) can be generated to test analog IC's with a small silicon area overhead.

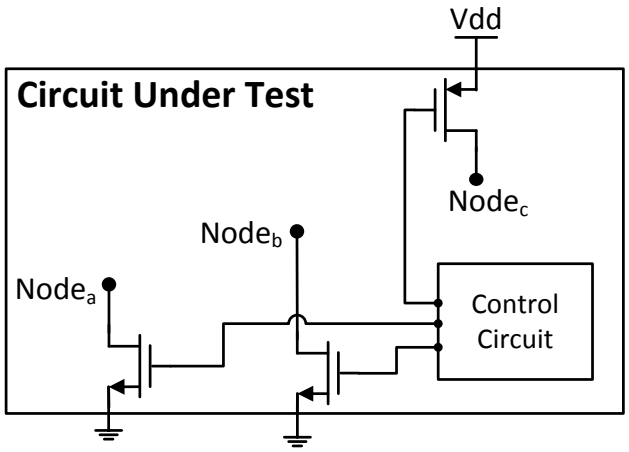


Figure 1: Block diagram representation of the proposed test infrastructure.

It is worth noting that no specific hypothesis is made about the considered circuit. However, both DfT techniques require extra-circuitry to be connected internally into the CUT. Therefore, the presented method is more suitable for low-frequency circuits such as the ones encountered in the automotive or biomedical industries. Circuits such as RF circuits can suffer from performance degradation due to the extra-parasitics introduced by the DfT blocks. In that case, the use of non-intrusive DfT techniques is advised and has been proposed in literature [13].

In the scope this work, the consideration of the sensitive nodes has been simplified by allowing the designers to make a pre-selection of the circuit nodes. The sensitive nodes of the circuit are therefore removed from the nodes to be considered and are not used by the two DfT methods. In the future, the analysis of the effects of the added parasitics will be automated and included in the workflow.

3.1. Topology Modification

The Topology Modification method consists in reconfiguring the targeted CUT to make defects observable. In [14], the modification of the value of some circuits components is used as a form of reconfiguration. In the scope of this work, the topology of the CUT is modified by means of small transistors added to the original circuit. These transistors are either connected between a node of the circuit and the ground (pull-down transistor), or a node of the circuit and the voltage supply (pull-up transistor), as illustrated in Figure 1. In the following, when the distinction between a pull-down (PD) and pull-up (PU) transistor is of no importance, the general denomination PX is adopted. The insertion of the PX transistors in the origi-

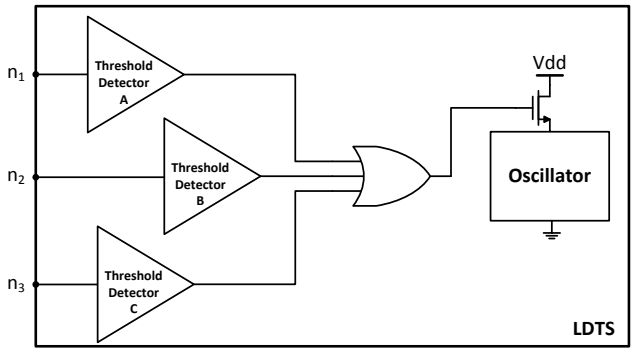


Figure 2: Internal organization of a LDTS block [11].

nal circuit C_0 leads to the set of topologies $\{T_0,T_1,...,T_p\}$, where T_0 is the original circuit C_0.

During the generation of these alternative topologies, two aspects have to be taken into consideration: the sizing of the used PX transistors and the stress imposed to the circuit when these are activated. After trials on the case studies from Section 5, it has appeared that minimal sizing for the PX transistors delivers convincing results. For what concerns the induced stress, the modified topologies are simulated and it is verified that they operate in the safe operating area (SOA) defined for the technology. The topologies which lead to excessive currents flowing through circuit branches are removed from the list of usable topologies.

During the normal operation of the circuit, the PXs are deactivated and hence do not have any effect on the circuit, besides the small capacitive parasitics inherent to their presence. During the test mode, these PX transistors are individually activated in order to make the CUT adopt different topologies. In the scope of this work, these modified topologies emerge from the activation of only one PX element at a time. The simultaneous activation of multiple PXs could be considered and would lead to more compact and more efficient solutions. However, this benefit would come at the expense of an increase in simulation time. Indeed, in the presented version of the method, the simulation time evolves linearly with the number of circuit nodes i.e. a circuit with N_n nodes has $2N_n$ alternative topologies because a PU or a PD transistor can be connected to each node. If two PXs could be activated simultaneously, the simulation time would evolve quadratically with the number of circuit nodes i.e. a circuit with N_n nodes has $4N_n(N_n - 1)$ alternative topologies. While it is not implemented in this work, solutions are given in Section 4 in order to cope with this increase in complexity.

3.2. Local Detection and Transmission System

The Local Detection and Transmission System (LDTS) entails adding small DfT building blocks in the targeted CUT to enhance the observability of the circuit. As illustrated in Figure 2, these DfT blocks consist of a number of threshold detectors with an embedded threshold detection

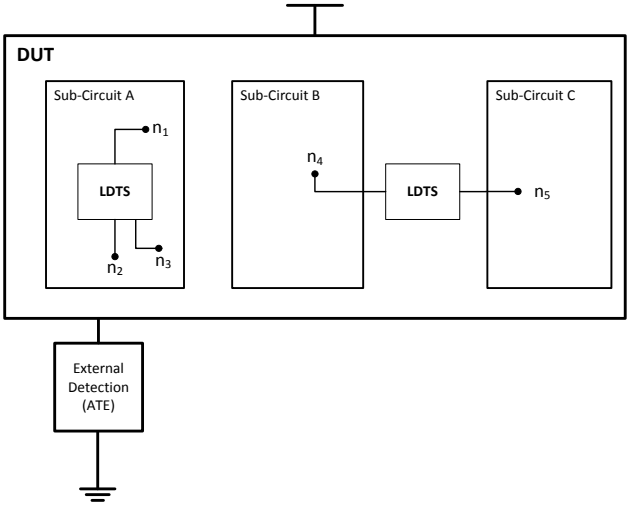


Figure 3: System-level view of the Local Detection and Transmission System [11].

voltage and an oscillator. These threshold detectors monitor the circuit nodes to which they are connected to verify that the selected node voltages are in their expected range. If a node voltage goes out of its expected range, the threshold detector is triggered and the oscillator is activated. The oscillator leaves a trace in the current consumption of the CUT which can be detected by the Automated Test Equipment (ATE) outside of the IC, as illustrated in Figure 3. Since the signal is carried in the IC's current consumption, no specific extra routing for the signal has to be designed and the local routing between the threshold detectors and the oscillator is sufficient.

The threshold detectors are autonomous in the sense that their detection threshold is embedded in their design. The proper sizing of their constituting transistors allows to define these embedded threshold voltages. This sizing is performed automatically by a simple optimizer taking the process variations into account.

By probing the node voltages and processing the information directly by the threshold detectors, the test infrastructures usually needed to bring node voltages outside the IC can be avoided. In this way, the required silicon area overhead is significantly reduced. Furthermore, the comparison of the node voltages to their expected range by means of the threshold detectors is operated in parallel. Therefore, the whole process of scanning the node voltages can be avoided and the test time can be reduced. It is noteworthy to say that this parallelized processing loses information which could be used for diagnosis purposes. In the scope of this paper, the focus is set on the fault detection rather than the fault diagnosis. Therefore, the minimization of the silicon area overhead was preferred to this information and the parallelization was chosen.

4. Test Structures Generation

Based on the DfT building blocks of the previous section, a procedure is proposed to combine the extra controllability and the extra observability in an optimal way. The final goal is to automatically generate the DfT test infrastructure for any given analog IC to maximize the fault coverage at minimum extra cost. The overall workflow can be decomposed into the set of tasks shown in the workflow diagram in Figure 4. In the following, these successive tasks are explained step by step. Thereafter, since the procedure involves two optimization steps, an overview about the used genetic algorithms and details over the underlying data structures are given.

4.1. Workflow of DfT generation

As shown in the workflow diagram from Figure 4, the procedure starts from the netlist of the circuit C_0 for which a test infrastructure has to be generated.

4.1.1. Lists generation

At first, three independent tasks take place in parallel: the *fault modeling*, the *nodes pre-selection* and the *topologies generation*. The fault modeling is the implementation of the defect-oriented method introduced in Section 2. Based on the netlist of the circuit C_0, the possible defects which can occur in the circuit are modeled by a list of Q faults $F_1, ..., F_Q$. The nodes pre-selection step consists in establishing a list of the R usable nodes $N_1, ..., N_R$ for the LDTS technique. This list is extracted as a sub-set of all the circuit nodes according to the restriction explained in Section 3. Finally, the topologies generation step consists in establishing the list of the S usable topologies $T_1, ..., T_S$ emerging from the Topology Modification technique. To that end, all the circuit nodes are first considered and two topologies are created for each of them i.e. one where the node is pulled down and one where the node is pulled up. Then, all the topologies failing to fulfill the restrictions explained in Section 3 are removed from the set.

Based on the lists of the Q faults, the list of the S alternative topologies and the list of the R usable nodes, the simulations can be carried out. The results of these simulations are used as input for the optimizer deciding which nodes have to be probed and in which topology so that the fault coverage is maximized and the test cost minimized. Furthermore, in order to be reliable, the proposed method has to take into account the effects of the process variations. However, if the method was applied in a straightforward way, a total of R*S*Q*N_c*N_{mc} simulations would be required, where N_c is the number of corners considered to assess the inter-die process variations and N_{mc} is the number of samples for each Monte-Carlo simulation. Hence, without making any adaptation, this number of simulations would explode quickly even for small circuits. In the following, refinements are presented in order to make the proposed DfT generation method tractable.

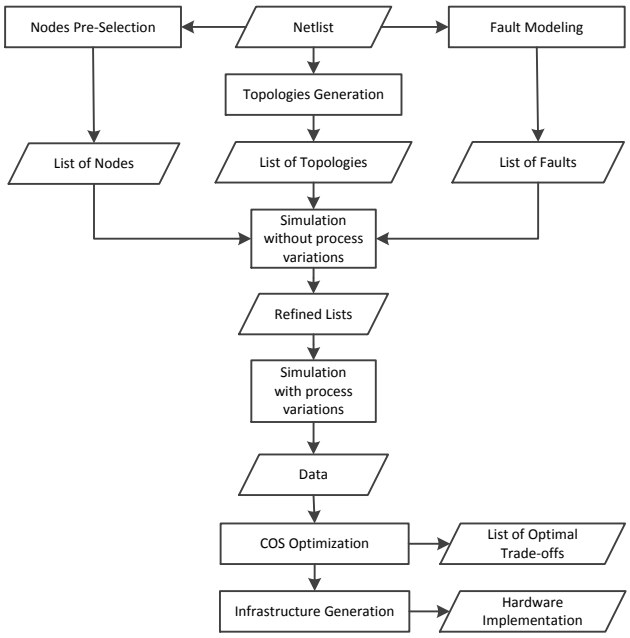


Figure 4: Workflow diagram of the automatic DfT generation method.

4.1.2. Lists refinement

The presented workflow proposes to first simulate the cases without considering the process variations. And, on the basis of these simulations, the cases where a fault does not produce a noticeable effect in a topology are put aside. In the scope of this work, a case (F_i, T_j, N_k) is removed from the list if:

$$|V_{ijk} - V_{0jk}| < 100mV$$

where V_{ijk} is the voltage at the node N_k in the topology T_j when the fault F_i is present, and V_{0jk} is the voltage at the node N_k in the topology T_j in the good circuit. The value of 100 mV is chosen according to the robustness of the used threshold detectors under process variations in the used technology.

This first selection allows to discard the case where a fault does not have a significant effect in the considered topology. The eliminated cases (F_i, T_j, N_k) do not have to be evaluated under process variations and this allows to reduce the computation time. Furthermore, this reduction in the number of (F_i, T_j, N_k) simplifies the optimization system which will be explained later.

It is worth pointing out that this list refinement step can be extended in order to cope with cases where the simulation time becomes an issue. This may arise when the considered circuit is relatively large or simply if the time allocated for the DfT generation is limited. In addition, a stronger refinement method would allow to make use of Topology Modification schemes evolving non-linearly with the number of circuit nodes, as introduced in Section 3.1 In such cases, a pre-selection of the usable topologies can be operated during the list refinement with a limited number of simulations. A way to make this pre-selection consists

in selecting only the alternative topologies which are sufficiently different from the original circuit. The difference between the original topology T_0 and an alternative topology T_i can be assessed by considering the voltages at each circuit node in both topologies and by calculating the difference. By summing up the differences for all the circuit nodes, a number can be built which summarizes the difference between the two topologies. Based on that quantified difference, a limited number of K topologies can be selected such that they present a maximum difference with the original circuit, where K can be chosen by the user. Another possibility consists in selecting the K topologies which maximize the difference with each other. In such cases, a total of R*K*Q*N_c*N_{mc} simulations are required and K can be adapted to fit a chosen time constrain. However, in the scope of this work, the exhaustive approach is chosen.

4.1.3. Data Generation

From this point, for the reduced set of (F_i, T_j, N_k), the simulations are carried out in the presence of process variations and their effects are evaluated. This evaluation of the variability is an important phase for the reliability of the method. It allows to choose the threshold detectors and guarantee the validity of the generated infrastructures across the process variations. To that end, the variability has to be evaluated for the CUT and for the threshold detectors which have to be added into the circuit. These two evaluations are carried out separately. The processing of these tolerance margins is carried out during the optimization step.

For the circuit under test, the inter-die process variations are simulated by a 4-sigma corner analysis. The intra-die variability is assessed by a 100-sample Monte-Carlo simulation at the typical corner. The assumption is made that this mismatch is the same for the other corners. Consequently, a total number of R*S*Q*(N_{mc}+N_c) simulations are finally required instead of R*S*Q*N_{mc}*N_c simulations.

For the threshold detectors, no actual simulation can be carried out at this stage of the method since these blocks are not yet generated. Instead of using a large library of threshold detectors, the method proposes to automatically generate them at the final stage of the method. Therefore, a model has been built for the used technology and the architectures utilized by the LDTS technique i.e. inverters or improved inverters [11]. This model is built based on a set of designed threshold detectors and predicts the process variations for a given detection voltage.

4.1.4. Controllability and Observability co-optimization

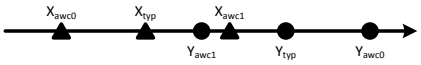
Based on the dataset resulting from the simulations, the topologies and nodes to be probed are decide upon in an optimal way by an optimizer. This optimization system has to reflect the desired specifications of a test system i.e. a maximal fault coverage, a minimal extra silicon area and a minimal test time. In this work, these specifications are straightforwardly translated to the components of the infrastructure. The test time can be attributed to the number of topologies since these have to be sequentially activated. Hence, the test time is proportional to the number of required topologies. Also, the silicon area required for the topology modification technique is negligible and the total silicon area can be approximated by the number of threshold detectors.

This simple translation gives rise to the following optimization system targeting three objective functions:

- a maximum fault coverage,
- a minimum number of topologies,
- a minimum number of threshold detectors.

As this optimization system is multi-objective, the solution is not unique but presents a Pareto-optimal set of solutions trading off fault coverage for silicon area and test time. To find this optimal set, an exhaustive search is not desirable. Indeed, there exist 2^R different combinations for the topologies selection, with R being the number of possible topologies. And if the problem of the threshold detectors is simplified to choosing between 10 discrete voltage levels, there exist 10^S possible cases, with S being the number of circuit nodes. Therefore, this combinatorial problem present a total of $2^R * 10^S$ cases which cannot be exhaustively searched. In this work, the multi-objective optimization was carried out by a genetic algorithm. The details of this optimization are developed in the next section dedicated to the implemented genetic algorithm and the data structures used.

Finally, one important point of the co-optimization is the processing of the process variations data simulated in the previous step. As introduced, when a threshold detector is added to the CUT to probe if the node voltage does not exit its expected range, the detection process must remain valid across the process variations. Therefore, design margins have to be taken into account during the node selection and the automatic design. However, while little can be done for mismatch, a refinement of these margins is done by considering the spatial correlation of the inter-die variability. Indeed, since the threshold detectors and the circuit are fabricated on the same die, they will have the same corner. This refinement is illustrated in Figure 5 where two cases are presented for the same six values from a fictive corner analysis on two features X and Y. In the first case, shown in Figure 5a, the two distributions overlap due to process variations since the value of X_{awc1} is smaller than the value Y_{awc1}. In the second case, shown in Figure 5b, the two distributions also seem to overlap. However, by definition, the corners awc0 and awc1 cannot occur at the same place in a manufactured integrated circuit, and the two distributions can actually be considered as separated. This refinement is taken into account in the optimization proces, and therefore the over-design of the threshold detectors is avoided.



(a) The two features X and Y overlap due to the process variations.

(b) The two features X and Y do not overlap.

Figure 5: Corner analysis of two features X and Y.

4.1.5. Generation of a hardware solution

The final step of this method consists in the designer or test engineer selecting a solution from the Pareto-optimal set proposed by the algorithm. Up to this point, the solution is a list of nodes which have to be controlled by the Topology Modification technique and a list of nodes which have to be monitored by the LDTS technique. The actual hardware implementation still has to be generated i.e. the generic DfT building blocks have to be sized and added into the circuit C_0.

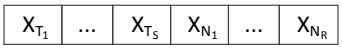
In the case of the Topology Modification technique, the selected pull-up and pull-down transistors are connected to the nodes of the circuit. Besides, in order to activate them during the test mode, a control circuit also needs to be generated. This control circuit is implemented by flip-flops connected in daisy-chain such as proposed in [9]. Additionally, the LDTS blocks also have to be generated according to the requirements provided by the selected solution. In the case of the oscillators and the OR-gates, the blocks are pre-designed and stored in a library. The threshold detectors, however, have to be generated on-demand according to specified detection voltages. This automatic design is carried out by the genetic algorithm explained in the next section.

4.2. Genetic Algorithms

Two important steps of the proposed DfT generation method rely on optimization systems. In the scope of this paper, these optimizations are carried out by a genetic algorithm. The implementation follows a non-dominated sorting multi-objective optimization scheme[15]. The core idea consists in starting from an initial random population representing a set of candidate solutions for the considered problem. Through modifications made to this population by genetic operations, the population evolves. Furthermore, by iteratively selecting the best candidates and removing the weakest according to a cost function, a global optimum emerges for this function. This evolution-based mechanism has been extensively covered in literature [16, 17]. In the following, the data structures supporting the implementation of the tackled problems are presented.

4.2.1. Controllability and Observability co-optimization

In the case of the selection of the topologies and the nodes to be probed, the idea consists in finding the optimal combinations of topologies and threshold detectors to use. Therefore, the data structure representing the candidates is a vector of variables:



where the binary variables X_{T_i} indicate whether the topology T_i is used or not. The variables X_{N_i} are variables indicating both whether a threshold detector is connected to the node N_i and the corresponding threshold voltage which should be used. The double information carried by these continuous variables X_{N_i} is made possible by organizing the domain of these variables into regions as follows:

$-V_{max} < X_{N_i} < -V_{min}$:
A threshold detector is connected to the node N_i and is triggered when the voltage is lower than $|X_{N_i}|$,

$-V_{min} < X_{N_i} < V_{min}$:
No threshold detector is connected to the node N_i,

$V_{min} < X_{N_i} < V_{max}$:
A threshold detector is connected to the node N_i and is triggered when the voltage is higher than $|X_{N_i}|$,

where V_{min} and V_{max} are the minimum and maximum detection threshold which can be implemented by the threshold detectors. These two values depend on the architecture adopted for the threshold detectors and the silicon technology which is used. In this work, these two values are equal to 0.8V and 2.5 V but this detection range can be extended by using more advanced threshold detectors.

Based on this data structure, the search for an optimal test infrastructure can be set as an (S+R)-dimensional optimization problem, where S is the number of possible topologies and R is the number of circuit nodes. This encoding scheme makes it possible for multiple topologies to use the same threshold detector but it allows only one threshold detector to be connected to a node. In order to connect two threshold detectors to the same node, the encoding can be changed to solution vectors of length S+2*R. This increase in complexity leads to a wider range of solutions. This scheme is applied to the second case study presented in Section 5.

The genetic algorithm evaluates a candidate solution by calculating the fault coverage corresponding to the infrastructure that it describes. The fault coverage is calculated based on the data generated during the simulations with the presence of process variations.

4.2.2. Hardware Generation

In the case of the automated generation of the hardware infrastructure, the threshold detectors have to be designed for the selected voltages, and the minimization of the effect of the process variations on this detection voltage is included as an objective. Such automated design methods already have been described in literature [18] and can

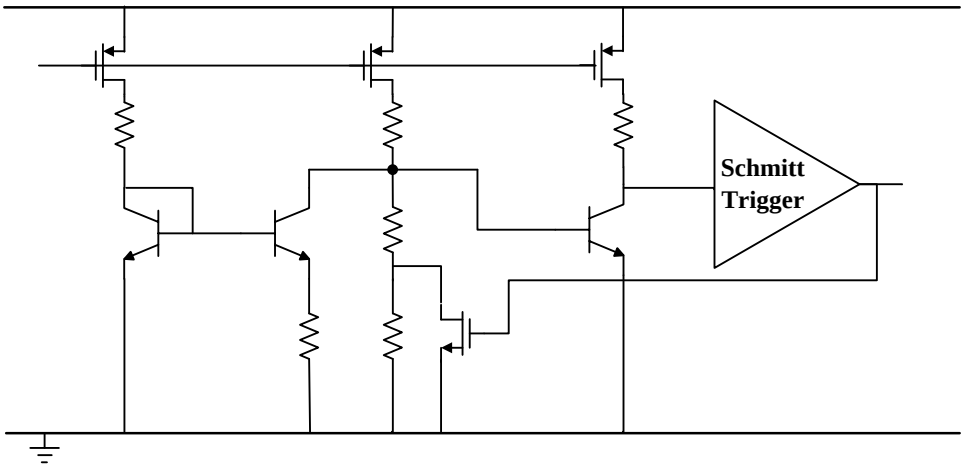


Figure 6: Schematic diagram of the Power-on-Reset circuit.

be carried out with genetic algorithms. In order to proceed to this automated design, the candidate solutions are represented by vectors:

W_1	L_1	...	W_N	L_N

where the W_i and L_i are the widths and lengths of the transistors constituting the threshold detectors presented in [9]. In the case where a simple inverter is used, only two transistors are involved and four variables are needed to describe a threshold detector. The design goals are translated as two objective functions minimizing the error on the detection threshold and minimizing the effects of the process variations.

5. Simulation Results

The proposed method has been applied to two industrial circuits: a Power-on-Reset (POR) circuit and a voltage regulator circuit. Both circuits are designed in 0.35μm BCD technology [19] and are basic building blocks present in numerous automotive IC's. Because these two circuits do not present easily accessible inputs, their controllability is very limited. Therefore, they are good candidates for the application of the Topology Modification method. Also, the observability provided by these circuits is limited. The POR circuit is terminated by a Schmitt Trigger, which presents an inverter at its end. Because of this, most of the faulty behaviors are filtered out and can never be obtained from the primary output. In the case of the voltage regulator, the inherent feedback mechanism also masks out defects and limits their observability from the primary output. In the following, the POR circuit will be discussed briefly. Then, the voltage regulator circuit will receive a more detailed analysis.

The first circuit under consideration is the Power-on-Reset circuit shown in Figure 6. Based on the fault models introduced in Section 2, the 4 bipolar and 12 MOSFET transistors lead to a list of 68 faults. The successive steps of the presented methods have been applied to it and Table 1 shows different hardware solutions from the obtained Pareto-optimal set. A coverage of 91.2% can be reached with 4 threshold detectors and 5 PX elements. In order to evaluate the silicon cost of this solution, the different areas of the building blocks were calculated after design by the hardware generation step. After comparison with the original size of the POR circuit, it is estimated that an increase of 10% in silicon area is needed to achieve that fault coverage.

Table 1: First 5 solutions in the Pareto-optimal front for the POR circuit.

Fault Coverage (percent)	**Number of threshold detectors**	**Number of PXs**
91.2	4	5
89.7	3	4
88.2	4	3
86.8	2	3
83.8	1	3

The second case study is the voltage regulator circuit illustrated in Figure 7. The fault modeling for the 25 MOSFET transistors leads to a list of 101 faults after removal of the redundant faults. The fault coverage achieved by a conventional specification-based testing was estimated to be around 68%. The proposed DfT generation method was applied to this circuit in order to improve its fault coverage. From the initial 75 possible topologies, 74 alternative topologies and the original circuit, only 60 were considered due to the restrictions explained in Section 3. For the nodes to be probed no pre-selection was made in this case and the 37 circuit nodes were considered. After the refinement step, a total of 86519 (F_i, T_j, N_k)-cases were simulated with the presence of process variations and stored in the database as explained in Section 4. The simulation of these cases took 6 hours when carried out by a parallelized implementation on a workstation equipped with a 2.5GHz quad-core processor and 12 GB of RAM. In

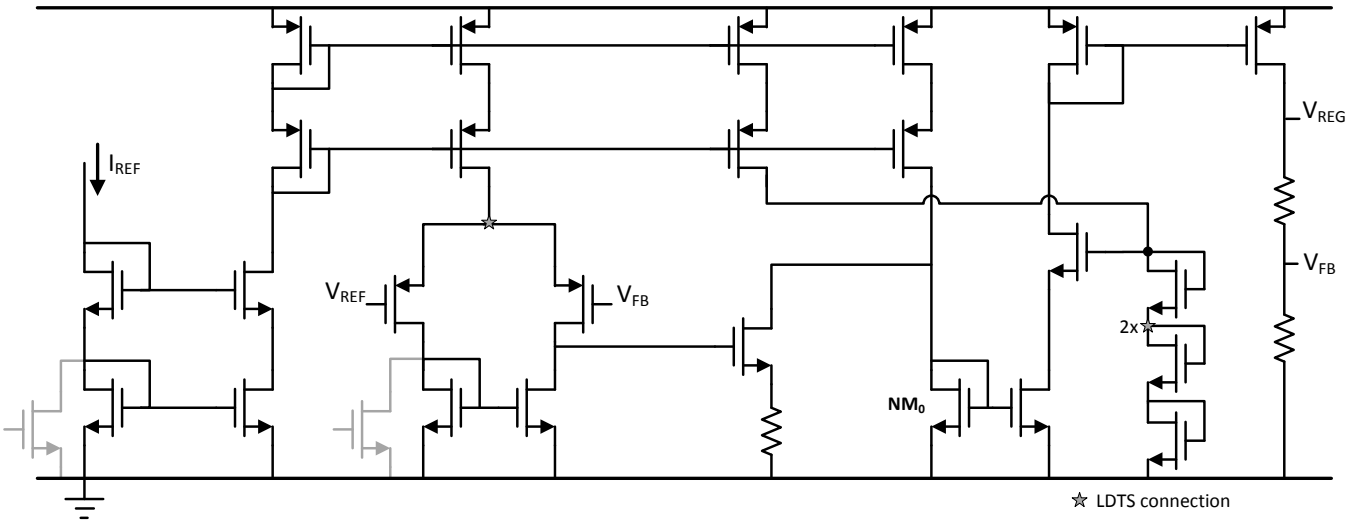


Figure 7: Schematic diagram of the voltage regulator circuit including a DfT infrastructure.

Table 2: First 5 solutions in the Pareto-optimal front for the voltage regulator circuit.

Fault Coverage (percent)	Number of threshold detectors	Number of PXs
94.06	3	7
94.06	2	8
93.07	3	6
93.07	2	7
92.08	2	6

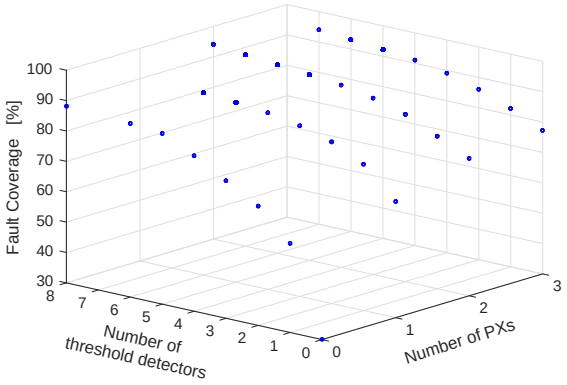


Figure 8: Scatter plot of the Pareto-optimal set of DfT solutions for the voltage regulator circuit.

comparison to this time-consuming task, the time taken by the other steps of the automated DfT generation method can be neglected.

Figure 8 shows the the Pareto-optimal set produced by the presented method from which five solutions have been taken and summarized in Table 2. Figure 7 illustrates a hardware solution achieving a 88.1% fault coverage. The generated infrastructure involves 2 PX transistors which are shown in gray and 3 nodes to be probed which are marked by stars. Two threshold detectors are connected to the same nodes. This was made possible by adopting solution vectors presenting S+2*R variables such as proposed in Section 4.2.1. The threshold detector probing the common mode node of the differential pair is triggered if the node voltage is above 1.85 V. The two threshold detectors probing the diode-connected transistors are triggered respectively if the node voltage is below 1.45 V or above 2.45 V.

In Table 2, it is shown that a maximum fault coverage of 94.1% can be reached with 7 threshold detectors and 3 PX elements. The analysis of the undetected faults shows that they originate from only two transistors in the circuit. One of these transistors is a transistor which is used as a capacitor on the feedback signal of the regulator and is not shown on schematic. When the drain or the source of this transistor is open, there is no noticeable effect on the circuit for the DC simulations. The other transistor is the one named NM_0 in Figure 7. The regulator feedback mechanism masks the effects of an open drain or an open source for the DC simulations.

Concerning the silicon area overhead, it is estimated that the maximum fault coverage of 94.1% can be achieved for an increase of less than 2% in silicon area. This significant difference with the first case study comes from the fact that the voltage regulator occupies six times more silicon area than the POR circuit. This difference in silicon area usage is due to the large transistors needed by the voltage regulator to drive the current supplying the integrated circuit.

Finally, it is worth noting that the results given for these two case studies are provided for an approach without fine selection of the test stimuli. The two circuits are supplied with signals considered as their normal conditions i.e. the POR circuit is supplied by a 3.3V voltage source and the regulator voltage by a 12V voltage source. The optimization of the applied signals could lead to more ef-

ficient solutions. However, in the scope of this work, the Topology Modification method was used as only source of test stimulus. The co-optimization of test stimulus could be studied as an extension of the proposed workflow.

6. Conclusion

A method has been presented for the automated generation of DfT infrastructures in order to test a given analog circuit. This method inserts generic DfT blocks offering extra controllability and extra observability in the circuit under test. An algorithm has been developed to co-optimize this controllability and observability in an optimal way. By applying a defect-oriented approach, the flow of analog testing has been fully automated. As a result, a set of Pareto-optimal solutions trading off the fault coverage for the required silicon area is proposed to the test engineer. After selection of a solution, a hardware implementation is automatically generated.

The proposed method has been applied to two industrial circuits and it has been shown that a fault coverage of 94.1% can be obtained for a silicon area overhead of less than 10%.

Acknowledgment

This research has been carried out within the frame of the SAFE-IC project funded by the IWT (code: 120642).

References

[1] P. Goel, An implicit enumeration algorithm to generate tests for combinational logic circuits, Vol. 100, IEEE, 1981, pp. 215–222.

[2] H. Fujiwara, T. Shimono, On the acceleration of test generation algorithms, Vol. 100, IEEE, 1983, pp. 1137–1144.

[3] C.-L. Hsu, Y. Lai, S.-W. Wang, Built-in self-test for phase-locked loops, Vol. 54, IEEE, 2005, pp. 996–1002.

[4] G. Renaud, M. J. Barragan, S. Mir, Design of an on-chip stepwise ramp generator for adc static bist applications, in: Mixed-Signal Testing Workshop (IMSTW), 2015 20th International, IEEE, 2015, pp. 1–6.

[5] C.-L. Wey, Built-in self-test (BIST) structure for analog circuit fault diagnosis, Vol. 39, IEEE, 1990, pp. 517–521.

[6] L. T. Wurtz, Built-in self-test structure for mixed-mode circuits, Vol. 42, Institute of Electrical and Electronics Engineers, 1993, pp. 25–29.

[7] A. Coyette, B. Esen, R. Vanhooren, W. Dobbelaere, G. Gielen, Automatic generation of lightweight controllability and observability structures for analog circuits, in: Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2015 International Conference on, IEEE, 2015, pp. 1–4.

[8] B. Kruseman et al., Defect oriented testing for analog/mixed-signal devices, in: Test Conference (ITC), 2011 IEEE International, IEEE, 2011, pp. 1–10.

[9] A. Coyette, B. Esen, W. Dobbelaere, R. Vanhooren, G. Gielen, Automated testing of mixed-signal integrated circuits by topology modification, in: VLSI Test Symposium, 2015.(VTS 2015). proceedingss 33th IEEE, IEEE, 2015.

[10] S. Sunter, K. Jurga, P. Dingenen, R. Vanhooren, Practical random sampling of potential defects for analog fault simulation, in: Test Conference (ITC), 2014 IEEE International, 2014, pp. 1–10.

[11] A. Coyette, B. Esen, R. Vanhooren, W. Dobbelaere, G. Gielen, Automatic generation of autonomous built-in observability structures for analog circuits, in: Test Symposium (ETS), 2015 20th IEEE European, 2015.

[12] M. Soma, S. Huynh, J. Zhang, S. Kim, G. Devarayanadurg, Hierarchical atpg for analog circuits and systems, Vol. 18, 2001, pp. 72–81.

[13] L. Abdallah, H. G. Stratigopoulos, S. Mir, C. Kelma, Experiences with non-intrusive sensors for rf built-in test, in: Test Conference (ITC), 2012 IEEE International, 2012, pp. 1–8.

[14] G. Srinivasan, S. Goyal, A. Chatterjee, Reconfiguration for enhanced alternate test (realtest) of analog circuits, in: Test Symposium, 2004. 13th Asian, 2004, pp. 302–307.

[15] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast elitist multi-objective genetic algorithm: NSGA-II, Vol. 6, 2000, pp. 182–197.

[16] M. Gen, R. Cheng, Genetic algorithms and engineering optimization, Vol. 7, John Wiley & Sons, 2000.

[17] V. M. Kureichik, S. P. Malioukov, V. V. Kureichik, A. S. Malioukov, Genetic algorithms for applied CAD problems, Vol. 212, Springer Science & Business Media, 2009.

[18] M. Chu, D. J. Allstot, Elitist nondominated sorting genetic algorithm based rf ic optimizer, Vol. 52, IEEE, 2005, pp. 535–545.

[19] F. De Pestel et al., Deep trench isolation for a 50V 0.35um based smart power technology, in: European Solid-State Device Research, 2003. ESSDERC'03. 33rd Conference on, IEEE, 2003, pp. 191–194.